AMENDMENTS TO THE CLAIMS

Please amend claim 42 as follows.

1. (Original) In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving a first input, the first input specifying a first behavior, the first behavior indicating how to change a value of a first parameter of the object over time;

animating the object by changing the value of the first parameter of the object over time according to the specified behavior; and

outputting the animated object.

2. (Original) The method of claim 1, wherein the object comprises a two-dimensional object.

3. (Original) The method of claim 1, further comprising receiving a second input, the second input specifying a keyframc indicating the value for the first parameter of the object at a first point in time, and wherein animating the object comprises changing the value of the first parameter of the object according to the specified behavior and further according to the specified keyframe.

4. (Original) The method of claim 1, further comprising receiving a second input, the second input specifying a second behavior, the second behavior indicating how to change a value of a second parameter of the object over time, and wherein animating the object further comprises changing the value of the second parameter of the object according to the second specified behavior.

5. (Original) The method of claim 1, further comprising receiving a second input, the second input specifying a second behavior, the second behavior indicating how to change the value of the first parameter of the object over time, and wherein animating the object comprises changing

the value of the first parameter of the object according to the first specified behavior and the second specified behavior.

6. (Original) The method of claim 5, wherein changing the value of the first parameter of the object according to the first specified behavior and the second specified behavior comprises determining a combined change to the value of the first parameter of the object according to a combination of the first specified behavior and the second specified behavior.

7. (Original) The method of claim 1, wherein the first behavior comprises one from a group consisting of:

a Fade In/Fade Out behavior;

a Grow/Shrink behavior;

a Motion Path behavior;

a Snap Alignment to Motion behavior;

a Spin behavior;

a Throw behavior;

an Align to Motion behavior;

an Attracted To behavior;

an Attractor behavior;

a Drag behavior;

a Drift Attracted To behavior;

a Drift Attractor behavior;

an Edge Collision behavior;

a Gravity behavior;

an Orbit Around behavior;

a Random Motion behavior;

a Repel behavior;

a Repel From behavior;

a Rotational Drag behavior;

a Spring behavior;

a Vortex behavior; and

a Wind behavior.

8. (Original) The method of claim 1, wherein the object comprises a text object and the first behavior comprises one from a group consisting of:

a Crawl Left behavior;

a Crawl Right behavior;

a Scroll Up behavior;

a Scroll Down behavior;

a Randomize behavior;

a Sequence behavior;

a Position behavior;

a Rotation behavior;

an Opacity behavior;

a Scale behavior;

a Tracking behavior; and

a Type On behavior.

9. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should be averaged over time.

10. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should be changed using a user-specified custom change.

11. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should be negated.

12. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should oscillate over time.

13. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should ramp over time.

14. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should be randomized.

15. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should change over time according to a specified rate.

16. (Original) The method of claim 1, wherein the first behavior indicates that changes to the value of the first parameter of the object should be executed in reverse order.

17. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should not change.

18. (Original) The method of claim 1, wherein the first behavior indicates that the value of the first parameter of the object should wriggle over time.

19. (Original) The method of claim 1, wherein the object comprises one from a group consisting of:

an image object;

a text object;

a particle system;

a filter;

a generator; and

a behavior.

20. (Original) The method of claim 1, wherein the first behavior comprises at least one user-settable behavior parameter, the method further comprising receiving a second input specifying a value for the behavior parameter, and wherein animating the object comprises changing the value of the first parameter of the object according to the first specified behavior and the specified value for the behavior parameter.

21. (Original) In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving an input, the input specifying the object;

creating one or more duplicates of the object according to a first plurality of parameters; and

animating the one or more duplicates by changing a value of a parameter of a duplicate over time according to a second plurality of parameters.

22. (Original) The method of claim 21, further comprising receiving an input, the input specifying a parameter, and wherein animating the one or more duplicates comprises changing the value of the parameter of the duplicate over time according to the second plurality of parameters and the specified parameter.

23. (Original) A user interface for a computer program for animating an object according to a behavior, the behavior having at least one user-settable parameter specifying how the behavior changes a value of at least one parameter of the object, the user interface comprising:

a control area; and

a user-manipulable control element located within the control area, for specifying a value for the at least one user-settable parameter of the behavior.

24. (Original) The user interface of claim 23, wherein the user-manipulable control element comprises a representation of a vector having a magnitude and an orientation.

25. (Original) The user interface of claim 24, wherein the control area comprises a circle, and wherein the representation of the vector comprises an arrow, the body of the arrow being a straight line, the tail of the arrow located in the center of the circle, the arrow pointing toward a point on the circumference of the circle.

26. (Original) The user interface of claim 25, wherein the magnitude of the vector controls a first user-settable parameter of the behavior, and wherein the orientation of the vector controls a second user-settable parameter of the behavior.

27. (Original) The user interface of claim 26, wherein the first user-settable parameter comprises a speed with which the object moves and wherein the second user-settable parameter comprises a direction in which the object moves.

28. (Original) The user interface of claim 27, wherein the behavior comprises a Throw behavior.

29. (Original) The user interface of claim 27, wherein the behavior comprises a Wind behavior.

30. (Original) The user interface of claim 23, wherein the user-manipulable control element comprises an arrow having a curved body.

31. (Original) The user interface of claim 30, wherein the control area comprises a circle, and wherein the curved body of the arrow comprises an arc of the circumference of the circle, the tail and the head of the arrow located on the circumference of the circle, the arrow pointing along the circumference of the circle.

32. (Original) The user interface of claim 31, wherein the length of the arrow controls a first user-settable parameter of the behavior, and wherein the direction of the arrow controls a second user-settable parameter of the behavior.

33. (Original) The user interface of claim 32, wherein the first user-settable parameter comprises a speed with which the object rotates and wherein the second user-settable parameter comprises a direction in which the object rotates.

34. (Original) The user interface of claim 33, wherein the behavior comprises a Spin behavior.

35. (Original) The user interface of claim 23, wherein the user-manipulable control element comprises.a first rectangle.

36. (Original) The user interface of claim 35, wherein the control area comprises a second rectangle, the second rectangle indicating an original size of the object.

37. (Original) The user interface of claim 36, wherein a difference between a width of the first rectangle and a width of the second rectangle controls a first user-settable parameter of the behavior, and wherein a difference between a height of the first rectangle and a height of the second rectangle controls a second user-settable parameter of the behavior.

38. (Original) The user interface of claim 37, wherein the first user-settable parameter comprises a change in the object's width and the second user-settable parameter comprises a change in the object's height.

39. (Original) The user interface of claim 38, wherein the behavior comprises a Grow/Shrink behavior.

40. (Original) The user interface of claim 23, wherein the user-manipulable control element comprises a first triangular region and a second triangular region.

41. (Original) The user interface of claim 40, wherein the control area comprises an area separating the first triangular region and the second triangular region.

42. (Currently Amended) The user interface of claim ~~23~~ 40, wherein a width of the first triangular region controls a first user-settable parameter of the behavior, and wherein a width of the second triangular region controls a second user-settable parameter of the behavior.

43. (Original) The user interface of claim 42, wherein the first user-settable parameter comprises a fade-in time of the object and the second user-settable parameter comprises a fade-out time of the object.

44. (Original) The user interface of claim 43, wherein the behavior comprises a Fade In/Fade Out behavior.

45. (Original) The user interface of claim 23, wherein the control area is semi-transparent.

46. (Original) A user interface for a computer program for animating an object, wherein animating an object comprises creating one or more duplicates of the object and animating the one or more duplicates by changing a value of a parameter of the one or more duplicates over time, the user interface comprising:

a control area; and

one or more controls for setting one or more parameters of the animation.

47. (Original) The user interface of claim 46, wherein the user-manipulable control element comprises a representation of a vector having a magnitude and an orientation.

48. (Original) The user interface of claim 47, wherein the control area comprises a circle, and wherein the representation of the vector comprises an arrow, the body of the arrow being a straight line, the tail of the arrow located in the center of the circle, the arrow pointing toward a point on the circumference of the circle.

49. (Original) The user interface of claim 48, wherein the magnitude of the vector controls a first user-settable parameter of the animation, and wherein the orientation of the vector controls a second user-settable parameter of the animation.

50. (Original) The user interface of claim 49, wherein the first user-settable parameter comprises a speed with which the one or more duplicates moves and wherein the second user-settable parameter comprises a direction in which the one or more duplicates moves.

51. (Original) The user interface of claim 46, wherein the user-manipulable control element comprises two points.

52. (Original) The user interface of claim 47, wherein the control area comprises a circle, and wherein the two points are located on the circumference of the circle, and wherein the two points specify a segment of the circle.

53. (Original) The user interface of claim 48, wherein the size of the segment of the circle controls a first user-settable parameter of the animation, and wherein the position of the segment of the circle controls a second user-settable parameter of the animation.

54. (Original) The user interface of claim 49, wherein the first user-settable parameter comprises a size of a range in which the one or more duplicates moves and wherein the second user-settable parameter comprises a location of the range in which the one or more duplicates moves.

55. (Original) The user interface of claim 46, wherein the control area is semi-transparent.

56. (Original) A method for generating a frame of an object using behaviors, comprising:

determining a current state of the object;

traversing a data structure to identify behaviors affecting the object;

accumulating forces for the behaviors affecting the object; and

generating a frame of the object according to the accumulated forces.

57. (Original) The method of claim 56, further comprising determining an initial velocity for the object.

58. (Original) The method of claim 56, wherein at least one of the behaviors is a motion behavior.

59. (Original) The method of claim 56, wherein at least one of the behaviors is a simulation behavior.

60. (Original) The method of claim 56, wherein at least one of the behaviors is a parameter behavior.

61. (Original) The method of claim 56, wherein the data structure comprises a tree structure.

62. (Original) The method of claim 56, wherein generating the frame comprises applying a mid-point method differential solver to determine a new parameter value for the object.

63. (Original) The method of claim 56, wherein the parameter value comprises a position of the object.

64. (Original) The method of claim 56, further comprising:

traversing the data structure to identify collisions; and

responsive to the existence of a collision, adjusting a system state to maintain a collision constraint.

65. (Original) The method of claim 56, further comprising iteratively repeating the animating step until a desired frame is reached.

66. (Original) The method of claim 56, wherein at least one object state is specified in terms of a keyframe, the method further comprising converting at least one keyframe into a set of forces that, when applied to the object, approximate the motion represented by the keyframe.

67. (Original) The method of claim 66, wherein converting at least one keyframe into a set of forces comprises deriving a set of forces based on the velocity and acceleration at the keyframe.

68. (Original) A method for generating an animation for an object using behaviors, the animation comprising a plurality of frames, the method comprising:

for each frame:

determining a current state of the object;

traversing a data structure to identify behaviors affecting the object;

accumulating forces for the behaviors affecting the object;

generating a frame of the object according to the accumulated forces; and

outputting the generated frame.

69. (Original) The method of claim 68, wherein at least one of the determining, traversing, accumulating, generating and outputting steps for a first frame is performed concurrently with at least one of the determining, traversing, accumulating, generating and outputting steps for a second frame.

70. (Original) A method for animating an object using parameter behaviors, comprising:

traversing a stack of operations on a range of values;

responsive to a single behavior value being sufficient to evaluate all operations in the stack, passing the single behavior value to each operation in the stack; and

responsive to a single behavior value not being sufficient to evaluate all operations in the stack:

determining a range of input values to compute a requested output range; and passing the determined range of input values to each operation in the stack.

71. (Original) A method for animating an object using a behavior, comprising:

outputting an original animation for the object according to a first behavior;

concurrently with outputting the object animation, accepting user input; and

outputting an updated animation for the object according to the user input.

72. (Original) The method of claim 71, wherein the user input comprises a command for changing a value of a parameter of the behavior, and wherein outputting the updated animation comprises outputting the updated animation according to the changed value of the parameter.

73. (Original) The method of claim 71, wherein the user input comprises a command for applying a second behavior to the object and wherein outputting the updated animation comprises outputting the updated animation according to the first and second behaviors.

74. (Original) The method of claim 71, wherein outputting the updated animation is performed without interrupting the animation for the object.

75. (Original) The method of claim 71, wherein the updated animation reflects the changed value of the parameter in real-time.

76. (Original) The method of claim 71, wherein outputting the original animation and outputting the updated animation each comprise rendering a plurality of frames and caching the rendered frames.

77. (Original) The method of claim 71, wherein outputting the original animation and outputting the updated animation each comprise rendering each of a plurality of frames sequentially.

78. (Original) The method of claim 71, wherein outputting the original animation and outputting the updated animation each comprise rendering each of a plurality of frames sequentially by calculating a current frame based on a previous frame.

79. (Original) The method of claim 71, wherein outputting the original animation and outputting the updated animation each comprise rendering a plurality of frames and periodically caching a subset of the rendered frames in an interval cache.

80. (Original) The method of claim 71, wherein outputting the original animation and outputting the updated animation each comprise evaluating, by a first thread, a first subset of frames, and evaluating, by a second thread, a second subset of frames.

81. (Original) The method of claim 80, wherein the first subset and the second subset of frames each comprise alternate frames of the animation.

82. (Original) In a computer-implemented animation system, a method for animating an object, the method comprising:

receiving a first input, the first input specifying a first behavior, the first behavior indicating how to change a value of a parameter of the object over time;

using at least one of a pixel shader and a vertex shader to generate a plurality of frames of the object, according to the specified behavior; and

outputting the plurality of frames.